Notice of Allowability	Application No.	Applicant(s)	
	10/728,732	BERNARDIN ET AL.	
	Examiner	Art Unit	
	Khanh Dinh	2151	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address--

All claims being allowable, PROSECUTION ON THE MERITS IS (OR REMAINS) CLOSED in this application. If not included herewith (or previously mailed), a Notice of Allowance (PTOL-85) or other appropriate communication will be mailed in due course. **THIS NOTICE OF ALLOWABILITY IS NOT A GRANT OF PATENT RIGHTS.** This application is subject to withdrawal from issue at the initiative of the Office or upon petition by the applicant. See 37 CFR 1.313 and MPEP 1308.

1. ☒ This communication is responsive to *11/15/2005*.

2. ☒ The allowed claim(s) is/are *35-68*.

3. ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
   a) ☐ All b) ☐ Some* c) ☐ None of the:
      1. ☐ Certified copies of the priority documents have been received.
      2. ☐ Certified copies of the priority documents have been received in Application No. _____ .
      3. ☐ Copies of the certified copies of the priority documents have been received in this national stage application from the International Bureau (PCT Rule 17.2(a)).

   * Certified copies not received: _____.

Applicant has THREE MONTHS FROM THE "MAILING DATE" of this communication to file a reply complying with the requirements noted below. Failure to timely comply will result in ABANDONMENT of this application.
THIS THREE-MONTH PERIOD IS NOT EXTENDABLE.

4. ☐ A SUBSTITUTE OATH OR DECLARATION must be submitted. Note the attached EXAMINER'S AMENDMENT or NOTICE OF INFORMAL PATENT APPLICATION (PTO-152) which gives reason(s) why the oath or declaration is deficient.

5. ☐ CORRECTED DRAWINGS (as "replacement sheets") must be submitted.
   (a) ☐ including changes required by the Notice of Draftsperson's Patent Drawing Review (PTO-948) attached
      1) ☐ hereto or 2) ☐ to Paper No./Mail Date _____.
   (b) ☐ including changes required by the attached Examiner's Amendment / Comment or in the Office action of Paper No./Mail Date _____.

Identifying indicia such as the application number (see 37 CFR 1.84(c)) should be written on the drawings in the front (not the back) of each sheet. Replacement sheet(s) should be labeled as such in the header according to 37 CFR 1.121(d).

6. ☐ DEPOSIT OF and/or INFORMATION about the deposit of BIOLOGICAL MATERIAL must be submitted. Note the attached Examiner's comment regarding REQUIREMENT FOR THE DEPOSIT OF BIOLOGICAL MATERIAL.

Attachment(s)

1. ☒ Notice of References Cited (PTO-892)
2. ☐ Notice of Draftperson's Patent Drawing Review (PTO-948)
3. ☐ Information Disclosure Statements (PTO-1449 or PTO/SB/08), Paper No./Mail Date _____
4. ☐ Examiner's Comment Regarding Requirement for Deposit of Biological Material
5. ☐ Notice of Informal Patent Application (PTO-152)
6. ☐ Interview Summary (PTO-413), Paper No./Mail Date _____ .
7. ☒ Examiner's Amendment/Comment
8. ☒ Examiner's Statement of Reasons for Allowance
9. ☐ Other _____.

Khanh Dinh
Primary Examiner
A.U. 2151

EXAMINER'S AMENDMENT

1. An examiner's amendment to the record appears below. Should the changes and/or additions be unacceptable to applicant, an amendment may be filed as provided by 37 CFR 1.312. To ensure consideration of such an amendment, it MUST be submitted no later than the payment of the issue fee.

Authorization for this examiner's amendment was given in a telephone interview with David Garrod (the Undersigned Attorney, Reg. No.35,149) on 11/17/2005.

The application has been amended as follows:

IN THE TITLE

Please amend the title of this application to read as follows:

"SCORE-BASED SCHEDULING OF SERVICE REQUESTS IN A GRID SERVICES COMPUTING PLATFORM"

IN THE SPECIFICATION

Please replace the new abstract as follows:

ABSTRACT OF THE DISCLOSURE

A service-oriented framework allows client applications to services hosted on a distributed computing access computational grid. Services facilitate remote, parallel

execution of code in a way that is scalable, fault-tolerant, dynamic and language-independent. Services can be written in a variety of languages, and do not need to be compiled or linked with vendor-supplied code. A client written in one language can invoke a Service written in another. A benefit of the invention over traditional approaches is that it virtualizes the Service. Rather than send a request directly to the remote machine hosting the Service, a client request is sent to a manager, which enqueues until an Engine is available. The first Engine to dequeue the request hosts the Service. This mechanism, in which a single virtual Service instance (the client-side object) is implemented by one or more physical instances (Engine processes), provides for fault tolerance and essentially unlimited scalability.

IN THE CLAIMS

Kindly amend the claims to read as follows:

1. (canceled)

2. (canceled)

3. (canceled)

4. (canceled)

5. (canceled)

6. (canceled)

7. (canceled)

8. (canceled)

9. (canceled)

10. (canceled)

11. (canceled)

12. (canceled)

13. (canceled)

14. (canceled)

15. (canceled)

16. (canceled)

17. (canceled)

18. (canceled)

19. (canceled)

20. (canceled)

21. (canceled)

22. (canceled)

23. (canceled)

24. (canceled)

25. (canceled)

26. (canceled)

27. (canceled)

28. (canceled)

29. (canceled)

30. (canceled)

31. (canceled)

32. (canceled)

33. (canceled)

34. (canceled)

35. (new) A method for providing computational services to a client using a grid-based distributed computing system, the system including a plurality of engines and at least one grid manager, the method comprising:

deploying executable code corresponding one or more service(s) such that the engines can access the executable code;

registering the service(s) with the grid manager;

creating instance(s) of the service(s) invocable by the client; and,

using one or more of the instance(s) to invoke one or more of the registered service(s), wherein invoking a service comprises:

communicating a service request to the grid manager;

using the grid manager to assign the service request to an available engine by (i) obtaining neediness score(s) for pending service request(s), each neediness score based on at least (a) an amount of time that engines have spent running task(s) associated with the requested service and (b) an amount of time that the request for the service has spent waiting to be assigned, (ii) obtaining affinity scores reflecting the affinity of available engine(s) to requested service(s), and (iii) applying an adaptive scheduling algorithm to assign the service request to one or more available engine(s); and,

executing code corresponding to the requested service on at least one of the assigned engine(s).

36. (new) A method for providing computational services to a client using a grid-based distributed computing system, as defined in claim 35, wherein the adaptive scheduling algorithm select(s) from available engine(s) based, at least in part, on one or more of the neediness score(s).

37. (new) A method for providing computational services to a client using a grid-based distributed computing system, as defined in claim 35, wherein the adaptive scheduling

algorithm select(s) from available engine(s) based, at least in part, on one or more of the affinity score(s).

38. (new) A method for providing computational services to a client using a grid-based distributed computing system, as defined in claim 35, wherein the adaptive scheduling algorithm select(s) from available engine(s) based, at least in part, on one or more of the neediness score(s) and one or more of the affinity score(s).

39. (new) A method for providing computational services to a client using a grid-based distributed computing system, as defined in claim 35, wherein the adaptive scheduling algorithm selectively supports redundant scheduling.

40. (new) A method for providing computational services to a client using a grid-based distributed computing system, as defined in claim 35, wherein deploying executable code comprises storing executable code on a shared file system accessible to the engines.

41. (new) A method for providing computational services to a client using a grid-based distributed computing system, as defined in claim 35, wherein deploying executable code comprises using a file update mechanism provided by the grid manager to distribute executable code to the engines.

42. (new) A method for providing computational services to a client using a grid-based distributed computing system, as defined in claim 41, wherein using a file update mechanism to distribute the executable code to the engines comprises distributing executable code to the engines when the grid manager is idle.

43. (new) A method for providing computational services to a client using a grid-based distributed computing system, as defined in claim 35, wherein creating an instance of a service comprises invoking a service creation method.

44. (new) A method for providing computational services to a client using a grid-based distributed computing system, as defined in claim 35, wherein communicating a service request to the grid manager comprises invoking a synchronous invocation method.

45. (new) A method for providing computational services to a client using a grid-based distributed computing system, as defined in claim 35, wherein communicating a service request to the grid manager comprises invoking an asynchronous invocation method.

46. (new) A method for providing computational services to a client using a grid-based distributed computing system, as defined in claim 35, wherein communicating a service request to the grid manager comprises synchronously invoking a client-side proxy that corresponds to the service.

47. (new) A method for providing computational services to a client using a grid-based distributed computing system, as defined in claim 35, wherein communicating a service request to the grid manager comprises asynchronously invoking a client-side proxy that corresponds to the service.

48. (new) A method for providing computational services to a client using a grid-based distributed computing system, as defined in claim 35, wherein communicating a service request to the grid manager comprises communicating a synchronous service request via a client-side Web Services Description Language (wsdl) proxy.

49. (new) A method for providing computational services to a client using a grid-based distributed computing system, as defined in claim 35, wherein communicating a service request to the grid manager comprises communicating an asynchronous service request via a client-side wsdl proxy.

50. (new) A method for providing computational services to a client using a grid-based distributed computing system, as defined in claim 35, further comprising providing a fail-over grid manager.

51. (new) A method for providing computational services to a client using a grid-based distributed computing system, as defined in claim 35, wherein each affinity score is computed, at least in part, from information representing the amount of the requested service's state that is already present on the engine-in-question.

52. (new) A method for providing computational services to a client using a grid-based distributed computing system, as defined in claim 35, wherein requested service(s) are first ranked according to neediness score(s), then assigned to engine(s) according to affinity score(s).

53. (new) A method for providing computational services to a client using a grid-based distributed computing system, the system including a plurality of engines and at least one grid manager, the method comprising:

registering service(s) with the grid manager;

deploying executable code(s) corresponding to the registered service(s) using a file update mechanism to distribute the executable code(s) to the engines, wherein at least some of said executable codes(s) are distributed while the grid manager is idle;

creating instance(s) of the service(s) that permit invocation by client(s); and,

using one or more of the instance(s) to invoke one or more of the registered service(s), wherein invoking a service comprises:

communicating a service request to the grid manager;

using the grid manager to assign the service request to an available engine by (1) computing a neediness score for the service request based, at least in part, on (i) a priority weight for the requested service, (ii) an amount of time that engines have spent running task(s) associated with the requested service, and (iii) an amount of time that the request for the service has spent waiting to be assigned, and (2) applying an adaptive scheduling algorithm to assign the service request to one or more available engine(s); and,

executing code(s) corresponding to the requested service on at least one of the assigned engine(s).

54. (new) A method for providing computational services to a client using a grid-based distributed computing system, as defined in claim 53, wherein the adaptive scheduling algorithm selectively supports redundant scheduling.

55. (new) A method for providing computational services to a client using a grid-based distributed computing system, as defined in claim 53, further comprising deploying executable code corresponding to the registered service(s) on a shared file system accessible to the engines.

56. (new) A method for providing computational services to a client using a grid-based distributed computing system, as defined in claim 53, wherein said deploying of executable code occurs when the grid manager is idle.

57. (new) A method for providing computational services to a client using a grid-based distributed computing system, as defined in claim 53, wherein communicating a service request to the grid manager comprises synchronously invoking a client-side proxy that corresponds to the service.

58. (new) A method for providing computational services to a client using a grid-based distributed computing system, as defined in claim 53, wherein communicating a service request to the grid manager comprises asynchronously invoking a client-side proxy that corresponds to the service.

59. (new) A method for providing computational services to a client using a grid-based distributed computing system, as defined in claim 53, wherein communicating a service request to the grid manager comprises communicating a synchronous service request via a client-side wsdl proxy.

60. (new) A method for providing computational services to a client using a grid-based distributed computing system, as defined in claim 53, wherein communicating a service

request to the grid manager comprises communicating an asynchronous service request via a client-side wsdl proxy.

61. (new) A method for providing computational services to a client using a grid-based distributed computing system, as defined in claim 53, wherein requested service(s) are first ranked according to neediness score(s), then assigned to engine(s) according to affinity score(s).

62. (new) A method for providing computational services to a client using a grid-based distributed computing system, the system including a plurality of engines and at least one grid manager, the method comprising:

obtaining executable code corresponding one or more service(s);

registering the service(s);

providing client-side proxy(s) for the registered service(s); and,

using one or more of the proxy(s) to invoke one or more of the registered service(s), wherein invoking a service comprises:

communicating a service request via a corresponding client-side proxy;

assigning the service request to at least one available engine by (i) obtaining a neediness score for the service request, said neediness score based on at least (a) an amount of time that engines have spent running task(s) associated with the service and (b) an amount of time that the request for the service has spent waiting to be assigned,

(ii) obtaining affinity scores reflecting the affinity of available engine(s) to the requested service, and (iii) applying an adaptive scheduling algorithm to assign the service request to one or more available engine(s); and,

servicing the request on at least one of the assigned engine(s).

63. (new) A method for providing computational services to a client using a grid-based distributed computing system, as defined in claim 62, wherein the service request is assigned based, at least in part, on the neediness score and one or more of the affinity score(s).

64. (new) A method for providing computational services to a client using a grid-based distributed computing system, as defined in claim 62, wherein the executable code is obtained from a shared file system accessible to the engines.

65. (new) A method for providing computational services to a client using a grid-based distributed computing system, as defined in claim 62, wherein communicating a service request comprises invoking a synchronous invocation method.

66. (new) A method for providing computational services to a client using a grid-based distributed computing system, as defined in claim 62, wherein communicating a service request comprises invoking an asynchronous invocation method.

67. (new) A method for providing computational services to a client using a grid-based distributed computing system, as defined in claim 62, wherein communicating a service

request comprises communicating a synchronous service request via a client-side wsdl proxy.

68. (new) A method for providing computational services to a client using a grid-based distributed computing system, as defined in claim 62, wherein communicating a service request comprises communicating an asynchronous service request via a client-side wsdl proxy.

Allowable Subject Matter

2. Claims 35-68 are allowed.

3. The following is an examiner's statement of reasons for allowance:

The above mention claims are allowable over the prior art of record does not appear to each or render obvious the claimed limitations in combination with the specific added limitations as recited in independent claims and subsequent dependent claims. None of the cited prior art discloses or teaches a method for providing computational services to a client using a grid-based distributed computing system, the system including a plurality of engines and at least one grid manager, the method comprising a combination of: creating instance(s) of the service(s) invocable by the client; using one or more of the instance(s) to invoke one or more of the registered service(s), wherein invoking a service comprises the step of communicating a service request to the grid manager. The invention further implements the grid manager to assign the service request to an available engine by obtaining neediness score(s) for pending service

request(s), each neediness score based on at least an amount of time that engines have spent running task(s) associated with the requested service and an amount of time that the request for the service has spent waiting to be assigned, obtaining affinity scores reflecting the affinity of available engine(s) to requested service(s), and applying an adaptive scheduling algorithm to assign the service request to one or more available engine(s).

For independent claim 53, None of the cited prior art discloses or teaches a method for providing computational services to a client using a grid-based distributed computing system, the system including a plurality of engines and at least one grid manager, the method comprising a combination of: deploying executable code(s) corresponding to the registered service(s) using a file update mechanism to distribute the executable code(s) to the engines, wherein at least some of said executable codes(s) are distributed while the grid manager is idle; creating instance(s) of the service(s) that permit invocation by client(s); and using one or more of the instance(s) to invoke one or more of the registered service(s), wherein invoking a service comprises the step of communicating a service request to the grid manager and using the grid manager to assign the service request to an available engine by (1) computing a neediness score for the service request based, at least in part, on (i) a priority weight for the requested service, (ii) an amount of time that engines have spent running task(s) associated with the requested service, and (iii) an amount of time that the request for the service has

spent waiting to be assigned, and (2) applying an adaptive scheduling algorithm to assign the service request to one or more available engine(s).

Conclusion

4. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Khanh Dinh whose telephone number is (571) 272-3936. The examiner can normally be reached on Monday through Friday from 8:00 A.m. to 5:00 P.m.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Zarni Maung, can be reached on (571) 272-3939. The fax phone number for this group is (571) 273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Khanh Dinh
Primary Examiner
Art Unit 2151
11/26/2005